UNIVERSIDAD EAFIT

Aplicación de técnicas de clusterización para la clasificación de música Dance Electrónica

Palabras Clave: audio processing, Convolutional Neural Network, K-Means, VGG, metrics, spectrogram

TRABAJO DE GRADO

Autor:
Carlos Alberto Murillo Martínez
Cmurill5@eafit.edu.co

Director:
Juan David Martínez Vargas
Correo: jdmartinev@eafit.edu.co

Codirector:
Marco Alunno
malunno@eafit.edu.co

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA
ESCUELA DE INGENIERÍAS
MEDELLÍN
2022

RESUMEN

El procesamiento de audio es una de las tareas esenciales para un científico de datos, el análisis de audio tiene aplicación en áreas muy diversas de conocimiento, como lo son: medicina, telecomunicaciones, mejorar la calidad de sonido en producciones musicales, inclusive aplicaciones militares (filtrar audio sospechoso o terrorista).

Con este proyecto se pretende utilizar técnicas de agrupamiento *hard* (K-Means o KNN) y *soft* (fuzzy clustering) para clasificar canciones de entrada, utilizando diferentes métricas. Se utilizarán los métodos de clasificación para segmentar audios de entrada previamente procesados y de esta manera obtener una muestra de segmentos representativos de las canciones y de esta manera determinar su similaridad con otras canciones del mismo género.

Otra técnica que ha probado ser efectiva para la clasificación de audio son las redes neuronales convolucionales (CNN) y se han utilizado para un gran campo de acción; en el ámbito musical se ha utilizado para clasificar técnicas de golpeo de arco en violín [1] hasta la detección de posibles problemas cardiacos utilizando los sonidos de los latidos del corazón [2]. En este proyecto utilizaremos esta técnica hasta el punto de la extracción de características y luego utilizaremos técnicas clásicas de clasificación para determinar a qué grupo pertenece una sección de canción.

ABSTRACT

Audio processing is one of the essential tasks for a data scientist, and audio analysis has applications in a diverse range of fields, such as medicine, telecommunications, improving sound quality in music production, and even military applications (filtering suspicious or terrorist audio).

This project aims to use hard clustering techniques (such as k-means or k-nearest neighbor) and soft clustering techniques (such as fuzzy clustering) to classify input songs using different metrics. The classification methods will be used to segment previously processed input audios and obtain a sample of representative segments of the songs, determining their similarity with other songs of the same genre.

Another technique that has proven effective for audio classification is convolutional neural networks (CNNs), which have been used in a wide range of fields. In the music field, they have been used to classify violin bowing techniques [1] and even detect potential heart problems using heartbeat sounds [2]. In this project, we will use this technique up to the point of feature extraction, and then use classical classification techniques to determine which group a section of a song belongs to.

CONTENIDO

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Según la física el sonido es una vibración que se propaga con una onda según el medio. Esta representación puede ser muestreada utilizando diferentes medios electrónicos, que transforman estos en una señal de audio que es una representación del sonido, puede ser una señal de un discurso, una canción o cualquier tipo de sonido.

La transformación de sonidos que son de naturaleza continua a una señal de audio implica algún tipo de técnica de muestreo, para trasladarlas a un sistema de almacenamiento como un cd o un archivo de audio. El problema que intentamos resolver es que a partir de la serie de atributos que podemos obtener de la representación matricial de una señal de audio convertida a la cual se le extraen diferentes características, aplicar una métrica de distancia a un sonido para poder caracterizarlo según su parecido o *distancia* musical usando métricas para su clasificación.

Al utilizar canciones completas para seleccionar la muestra de audio se elimina la influencia del ruido externo de la muestra, como ruido de ambiente u otros tipos de externalidades al momento de producir una canción. Sin embargo, para simular ruido en la muestra de datos se agregan diferentes versiones de la misma canción, porque son canciones que presentan el mismo ritmo pero tonalidades una escala mayor o menor. Esto ayuda a determinar si la clasificación es efectiva para este género musical porque si se puede clasificar efectivamente cada sección de audio se podría detectar que secciones son más cercanas a otras y podríamos utilizar esta clasificación para poder mezclar canciones de acuerdo con su similaridad.

1.2. JUSTIFICACIÓN

La clasificación de audio y la noción de distancia entre canciones es un concepto que se puede aplicar en muchos campos del análisis de datos, en este proyecto se realiza una clasificación de un género musical específico para y poder saber si una sección de canción se puede relacionar con otra y ser de insumo para un proyecto más grande de creación de un dj automático [3].

Como se menciona al inicio del documento la aplicabilidad del análisis de audio trasciende el ámbito académico, es una de las bases de las tecnologías de speech2text, en el campo de la medicina se puede utilizar para medir probabilidad que una persona pueda sufrir un infarto de acuerdo con el ritmo de los latidos de su corazón [2].

La representación computacional de una señal de audio puede ser una matriz con coordenadas de tiempo y cada columna representa una característica del audio, como: frecuencia, bits por segundo, etc.

Este trabajo pretende aplicar una metodología robusta de clasificación buscando secciones de canciones que puedan estar relacionadas con otras, Utilizando la noción de distancia entre canciones (esta distancia puede ser modificada para examinar diferentes estrategias que se acomoden mejor a la forma de los datos), utilizando un Espectograma o variables clásicas de análisis de audio como lo son: los MFCC’s, amplitude envelope, etc. [4].

1.3. OBJETIVOS

1.3.1. Objetivo general

Comparar técnicas de clusterización duras y suaves (hard clustering y soft-clustering) para la clasificación de archivos de audio en música Dance Electrónica utilizando diferentes metodologías para la selección de características.

1.3.2. Objetivo específico

- Calcular clústeres utilizando diferentes métodos de clasificación la distancia entre los diferentes audios.
- Validar la clasificación de acuerdo con juicio de expertos.
- Lograr una caracterización de los archivos de audio.
- Evaluar los resultados de clasificación y elegir la que mejor funciona para la base presentada.

2. ESTADO DEL ARTE Y MARCO TEÓRICO

La clasificación de sonidos es un campo de estudio muy amplio debido a sus diferentes aplicaciones en múltiples campos. Existen dos metodologías de análisis de audio que consisten en: Un análisis clásico utilizando variables directamente del archivo de audio [5] y otro análisis que consiste en transformar el audio en imagen y luego utilizar una arquitectura CNN para la extracción de características [2].

Al trabajar con variables clásicas usualmente el conjunto de datos está enfocado en la resolución de problemas específicos y con conjuntos de datos pequeños [6], este problema es similar al que se aborda en este documento, dado que los conjuntos de datos no han sido previamente etiquetados y se requiere lograr una clasificación.

Un algoritmo muy utilizado para clasificar grupos de datos, de manera no supervisada es K-Means, debido a que no existe una clasificación previa de los datos, en este caso se debe dejar que la métrica (inicialmente euclídea) sea la que defina de acuerdo con las características seleccionadas cual sería la mejor aproximación.

Por este motivo en esta sección se explica la naturaleza de las características del sonido su representación como una imagen y los métodos por los que se pueden extraer características.

Entre las variables que podemos utilizar para entender el sonido según su nivel de abstracción tenemos las siguientes [4]:

Abstracción	Características	Descripción
Alto	Instrumentación, llave, acordes, melodía ritmo, tempo, género	Características abstractas que el humano es capaz de definir
Medio	Pitch and beat-related descriptors, MFCC's, fluctuation patterns	Características que tienen sentido desde un punto de vista perceptual, pero no son realmente entendibles por las personas
Bajo	Amplitude envelope, energy, spectral centroid, spectral flux, zero-crossing-rate	Las puede entender un computador

Para la identificación de audio es necesario tomar más cosas en consideración, por ejemplo: el modelo más utilizado en la actualidad y que mejor resultado ha dado es transformar el audio en una imagen conocida como un espectograma, luego se procesan las imágenes utilizando Redes Neuronales Convolucionadas (CNN o ConvNet) [7], este acercamiento es muy útil porque permite hallar características claves para la identificación del audio.

Para entender mejor lo que es un espectograma debemos definir lo que es el espectro. El sonido puede representarse como la suma de diferentes frecuencias, el espectro se define como representación de la señal sonora en términos de las frecuencias que la componen. En otras palabras, el espectro es la representación de la señal en el dominio de la frecuencia. La frecuencia más baja se conoce como la frecuencia fundamental y los múltiplos de la frecuencia fundamental se les conoce como harmónicos.

Entonces el espectograma es un gráfico que muestra la frecuencia de las distintas señales contra el tiempo, en otras palabras, es un gráfico de sus espectros contra el tiempo por lo que algunas veces incluso se muestra en tres dimensiones, como se muestra en la imagen:

Ilustración 1 - espectograma 3d **[8]**

Pero su versión más utilizada es en dos dimensiones:

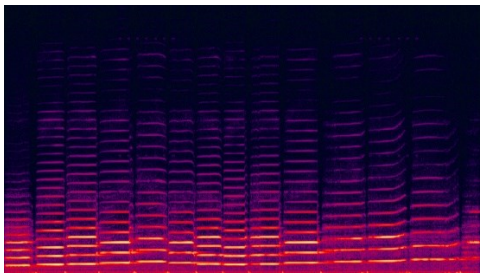


Ilustración 2 - espectograma audio de violín **[9]**

Conociendo que un espectograma muestra como varía la amplitud y la fase de cada frecuencia en el tiempo. se puede obtener una frecuencia media o una frecuencia mediana para obtener un patrón de un audio, en este punto entra la siguiente parte de la investigación.

Como se observa uno de los mayores problemas a los que se enfrenta utilizar los modelos de clasificación dura como K-Means, es la gran cantidad de datos que dispone para trabajar se deben simplificar muchas veces para analizarlos escoger, ¿Qué secciones del audio muestrear?, se debe pre procesar el audio para eliminar secciones de poca relevancia y por último determinar cuáles son los centroides con los que se debería trabajar [10].

Los algoritmos que se van a trabajar en este proyecto se aplican en un contexto de agrupamiento más no de identificación, la distancia entre dos sonidos puede tener muchos componentes de análisis distintos, en el que incluso la preparación de datos influye, por este motivo la arquitectura más utilizada para identificación de audio es basada en CNN.

Sin embargo, otra rama que es de mucha utilidad es la clasificación de audio, es analizar su MFCC (Mel-Frequency cepstral coefficients), como nos menciona [11] en su artículo de uso de KNN para clasificación de audio, *"El cepstrum de frecuencia mel ha demostrado ser muy eficaz en el reconocimiento de la estructura de las señales musicales y en el modelado del tono subjetivo contenido de frecuencia de las señales de audio"*.

Al aplicar esta técnica lo que se propone es realizar una serie de pasos para obtener características del audio, las cuales son principalmente (Ilustración 3) [12]:

- Separar la señal en pequeños tramos.
- A cada tramo aplicarle la Transformada de Fourier discreta y obtener la potencia espectral de la señal.
- Aplicar el banco de filtros correspondientes a la Escala Mel al espectro obtenido en el paso anterior y sumar las energías en cada uno de ellos.
- Tomar el logaritmo de todas las energías de cada frecuencia mel
- Aplicarle la transformada de coseno discreta a estos logaritmos.

Podemos aplicar técnicas para robustecer las medidas de distancia(inicialmente euclídea, pero se pueden evaluar otras durante el desarrollo del en los algoritmos que se van a trabajar, como aplicar

una normalización de los datos con el método MEL, normalizar la onda y eliminar secciones de amplitud baja (silencios).

En la investigación realizada por [13] en la cual utilizan "señales de audio y redes neuronales para la clasificación de resinas de plástico para reciclaje" utilizan varias de las técnicas mencionadas anteriormente, lo cual indica que podría ser una buena guía para la clasificación de música, .

2.1. MEL-FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

Es una técnica ampliamente utilizada para procesamiento de habla y sonidos del día a día. Esta técnica funciona calculando los coeficientes a partir del espectro de una pequeña señal de audio, obtenida utilizando una Transformación Rápida de Fourier (FFT), que toma en cuenta una señal muestreada en los dominios del tiempo y frecuencia. [13].

El procedimiento para obtener los diferentes MFCC puede ser representado de la siguiente manera:

Ilustración 3 - proceso obtención MFCC

Este proceso puede ser bueno para señales de audio pequeñas, pero para valores grandes de tiempo consume muchos recursos.

Para obtener un mejor entendimiento de lo que es un MFCC, hay que dividir el concepto en varias secciones [4]:

- **Escala de Mel**: Es una escala que permite realizar análisis de audio. Esta escala se basa en la capacidad de percepción humana y tiene a estar en el rango de 20 Hz hasta 20000Hz aunque no son valores fijos.
- **Coeficientes:** Son Valores que describen características del sonido
- **Cepstral:** Un adjetivo que proviene de *Cepstrum* que se relacionan 1 a 1 con los siguientes conceptos

Cepstrum	Spectrum
Quefrency	Frequency
Liftering	Filtering
Rhamonic	Harmonic

Tabla 1 - relación conceptos Cepstrum

Matemáticamente se define un Cepstrum con la siguiente fórmula:

$$C\big(x(t)\big) = F^{-1}[\log(F[x(t)])]$$

Donde:

→ $x(t)$ Señal de audio

→ F Transformación Discreta de Fourier

→ $F(x(t))$ Espectro

→ $\log(F[x(t)])$ Logaritmo de un Espectro

Es decir, que un cepstrum es aplicar la inversa al logaritmo de un spectrum.

Visualmente un cepstrum se visualiza de la siguiente manera, donde la primera gráfica representa una señal de audio en el dominio del tiempo, y la última gráfica es un Cepstrum:

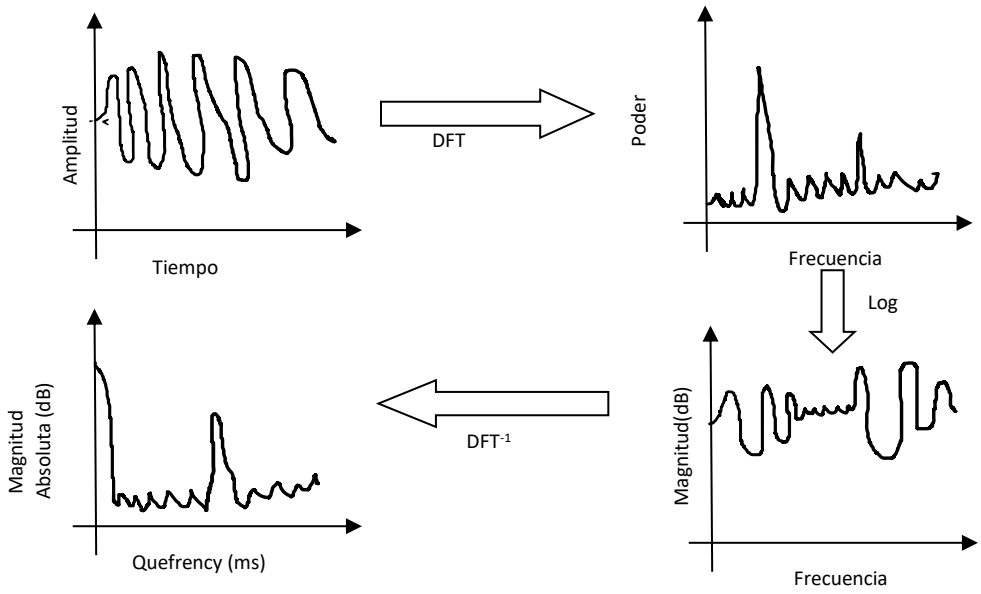


Ilustración 4 - proceso transformación Spectrum a Cepstrum

2.2. REDES NEURONALES CONVOLUCIONALES (CNN)

Las redes neuronales convolucionales son muy eficientes para resolver problemas de clasificación y asequibles comparados con otros modelos. [13]

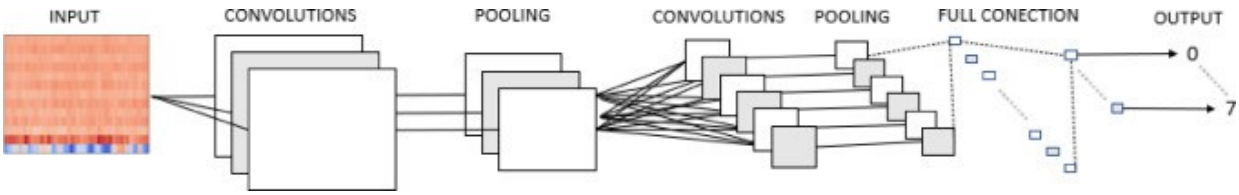


Ilustración 5 - Arquitectura básica CNN [13]

Esta arquitectura funciona tomando el insumo, que puede ser de cualquier naturaleza; imagen, audio, matriz. A partir de allí hay una capa de convolución que se encarga de extraer características de la entrada usando filtros. Le capa de *pooling* se encarga de reducir el tamaño de los datos para que una capa de convolución pueda obtener un mapa de características de los datos, así el modelo puede aprender de diferentes maneras por lo tanto se evita el sobreajuste [13].

Para realizar la clasificación se utiliza una función *softmax* en la que cada capa genera un número positivo, que en conjunto suman 1, estos valores pueden ser entendidos como probabilidades. Como se muestra en la ecuación:

$$softmax(x_i) = \exp(x_i) \,/\, \sum_{j=1}^{n} \exp\,(x_i)$$

Finalmente se utiliza una función de entropía que asigna los valores a cada categoría:

$$E(w) = \frac{1}{n}\sum_{i=1}^{n}[t_i.\log(y_i) + (1 - t_i).\log\,(1 - y_i)]$$

Donde *w* es la matriz de pesos, y_i el resultado deducido, t_i el resultado correcto y *n* la cantidad de objetos de muestra.

2.3. VISUAL GEOMETRY GROUP (VGG)

VGGNet es un tipo de Red neuronal convolucional que fue presentado por primera vez en el artículo *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION* [14], este modelo marca un hito en la clasificación de imágenes y desde que fue presentado ha ido adquiriendo nuevas versiones una de ellas se utiliza en este trabajo, para la extracción de características.

VGG19 es una red neuronal convolucional, derivada de la familia VGGNet, que consiste en 19 capas, 16 capas de convolución y 3 capas totalmente conectadas Este modelo alcanzó resultados de "estado del arte" con diferentes conjuntos de datos y ha sido muy aplicado para el procesamiento de imágenes [15].

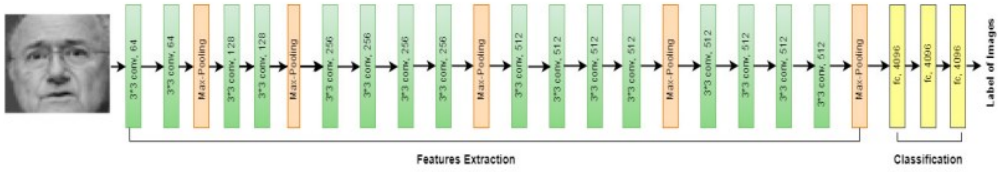


Ilustración 6 - Arquitectura VGG19

En este proyecto se aprovecha la representación del audio como una imagen (Ilustración **2**) para obtener características de esta, estas características son extraídas utilizando el procedimiento mostrado en la ilustración 6.

3. DATOS

3.1. PLAN DE GESTIÓN DE DATOS

Los archivos de audio suministrados de insumo fueron proporcionados por el profesor Marco Alunno en un contexto de continuar el proyecto desarrollado por uno de sus estudiantes de maestría, si existe la solicitud de crear copias de seguridad de estos, se almacenarán copias en distintos dispositivos.

Mientras que los datos de evaluación en caso de ser requeridos serán proporcionados por el profesor Marco Alunno.

La evaluación de los datos para poder determinar el mejor ajuste de la clasificación será realizada en principio por tres personas y la evaluación e los resultados por alguna muestra cualitativa de estudiantes de música.

Los datos derivados de este proyecto son públicos, sin embargo, las canciones seguirán siendo protegidas por los derechos de autor y son propiedad de sus creadores.

3.2. ADQUISICIÓN DE DATOS

Los datos de insumo para el proyecto fueron transferidos vía repositorio, una vez descargados de los datos que pertenecen a **53 canciones de género electrónica y dance**. Son almacenadas en un pc de trabajo local.

Todas las canciones tienen formato (.WAV) que indica una mejor calidad de sonido además de no ser comprimido, en comparación a otros formatos como (.mp3).

3.3. DESCRIPCIÓN Y ANÁLISIS PRELIMINAR DE LOS DATOS

Son 53 canciones del mismo género, la canción más larga tiene una duración de **10 minutos** y la más corta una duración de **3:56**. Todas tienen una frecuencia de muestreo de 44.1 kHz, estereo con muestras de 16 bits y una velocidad de transmisión de 1411Kbps (Kilobits por segundo) que es una velocidad de transmisión de datos buena para archivos de audio que indican buena calidad del archivo.

3.4. PREPROCESAMIENTO DE LOS DATOS

Esta sección será explicada en detalle de acuerdo con el modelo seleccionado. Sin embargo, el primer paso del procesamiento del audio será la separación del audio en **secciones iguales y comparables**, el método para lograr esa separación se estudia de acuerdo con el modelo seleccionado.

3.5. ASPECTOS ÉTICOS

Los archivos de audio que presenta el proyecto son canciones licenciadas para no violar leyes de copyright y no serán expuestas de manera pública (internet).

¿cuáles son los beneficios y quién se beneficiará?

Este trabajo permite analizar el primer paso dentro de la clasificación de archivos de audio y permitirá seleccionar una metodología que permita utilizar un mejor acercamiento en la música Dance.

4. DESARROLLO DE MODELOS

4.1. METODOLOGÍA

Para este trabajo se va a utilizar la metodología CRISP-DM:
"CRISP-DM son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos.

- *Como metodología, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas.*
- *Como modelo de proceso, CRISP-DM ofrece un resumen del ciclo vital de minería de datos."* (IBM Knowledge center, 2021)

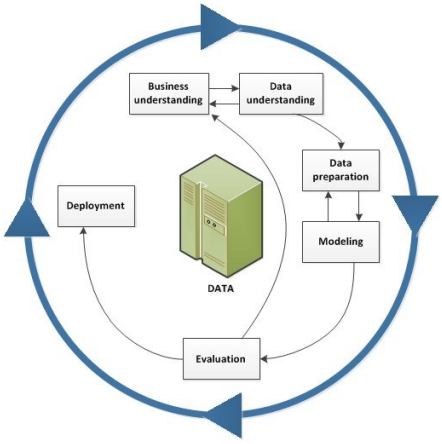


Ilustración 7 - Metodología CRISP-DM

Aplicado al proyecto se realizará de la siguiente manera:

- **Entendimiento del negocio:** En esta fase analizaremos el proyecto realizado por Víctor Tideman [3] para la obtención de la información. Además, se presenta un resumen de las decisiones que se tomaron a nivel general para el procesamiento del audio

- **Entendimiento de los datos:** En esta sección, se trata la descripción de los datos, cuantas variables y el tamaño del archivo. Finaliza explicando, cuál es el procedimiento para crear y seleccionar las variables necesarias para el modelo.

- **Preparación de datos:** Para realizar el análisis de audio se extraen características del archivo.

- **Modelado:** Aplicar los algoritmos de clasificación para determinar *distancia* entre canciones.

- **Evaluación:** Evaluar estos resultados aplicando los resultados del modelo sobre un conjunto de evaluación, algunas de las métricas que se plantean son: RMSE, Silhouette Analysis, cohesión (**juicio de expertos**).

- **Despliegue:** Desplegar los resultados en una máquina EC2 de AWS, un tablero de control en el cual se puede explorar de manera interactiva los resultados.

A continuación, se aplicará esta metodología para describir cada procedimiento de segmentación, utilizando un procedimiento clásico y otro donde las características se seleccionan utilizando una red neuronal.

4.2. FLUJO DE TRABAJO

Aplicando la metodología CRISP-DM el flujo de trabajo de todo el proyecto fue el siguiente:

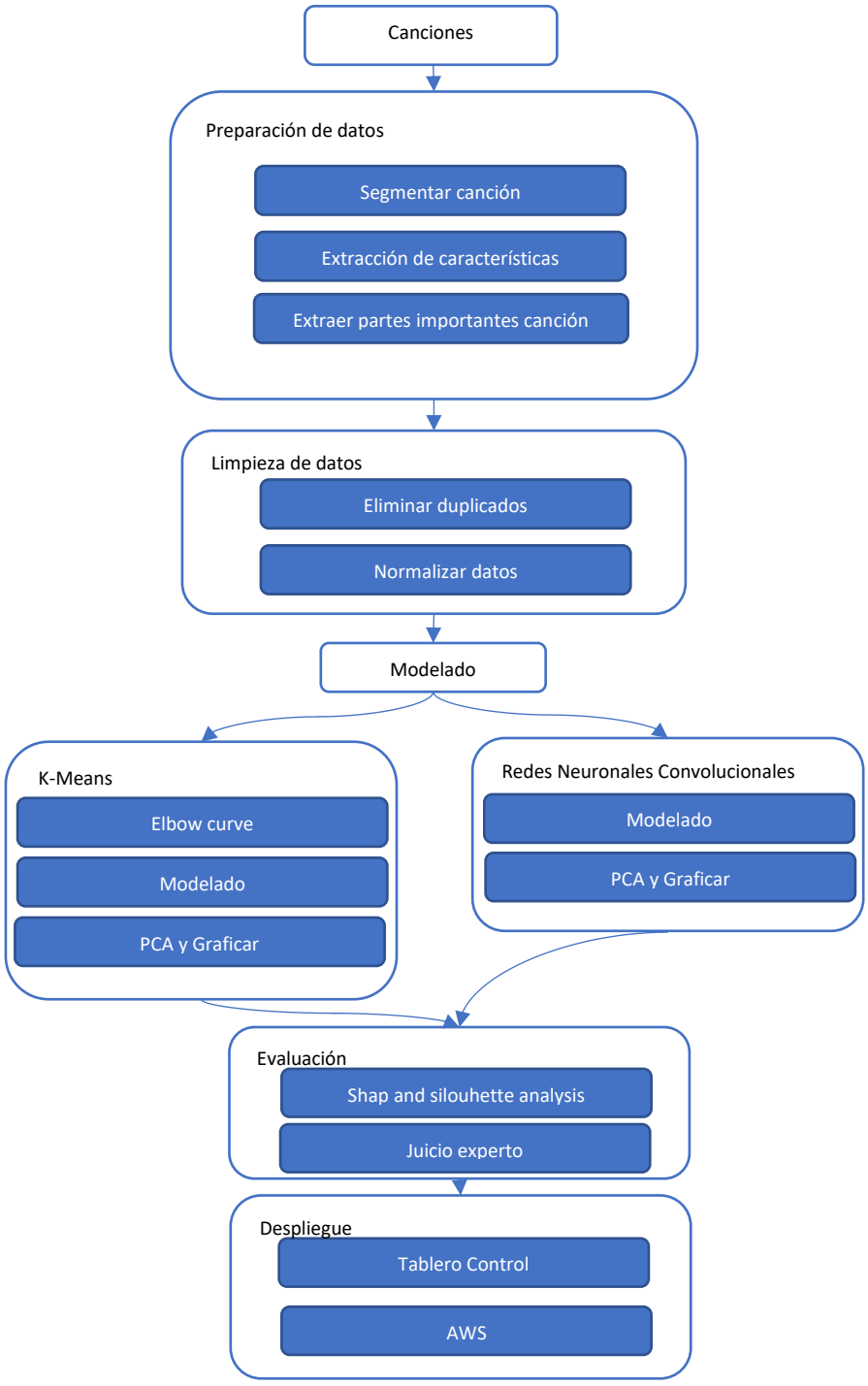


Ilustración 8 - Flujo de trabajo

4.3. PROCEDIMIENTO DE CLASIFICACIÓN UTILIZANDO VARIABLES CLÁSICAS

Explica el procedimiento realizado en el Jupyter Notebook: ***Clasificacion_generos_musicales.ipynb***.

4.3.1. Entendimiento de negocio

Uno de los puntos más importantes para definir el éxito o el fracaso de este acercamiento, será poder determinar que, si dos secciones de una canción son "cercanas" de acuerdo con una métrica euclídea, estas podrán ser mezcladas de manera consecutiva por un dj.

Para analizar un segmento de audio en un computador podemos utilizar segmentos mucho menores de un segundo, en la escala de milisegundos; pero, para que una persona pueda determinar similaridad entre canciones debe examinar un segmento algo más extenso, en la escala de segundos.

Por este motivo se toma una decisión de particionar las canciones en segmentos de **15 segundos** este valor es arbitrario, pero puede ser parametrizable para el proceso.

4.3.2. Entendimiento de los datos

Para realizar el análisis de los datos se utiliza la librería de Python librosa[1]. Con el apoyo de las librerías Pandas y Numpy. Para el trabajo con matrices y series.

Utilizando esta librería podemos transformar el formato .WAV en una matriz de una columna que esta en el rango de -1 y 1:

$$x(t) = \begin{bmatrix} x_i \\ \vdots \\ x_n \end{bmatrix} \; Donde \; x_x \in [-1, 1]$$

Gráficamente es una señal de audio de ejemplo:

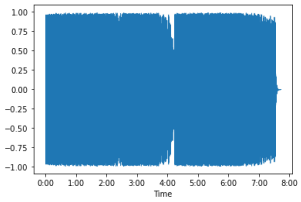


Ilustración 9 - Señal audio canción completa

Esta misma librería nos permite obtener características del sonido, como los mfcc's u obtener espectograma de la canción:

[1] https://librosa.org/doc/latest/index.html

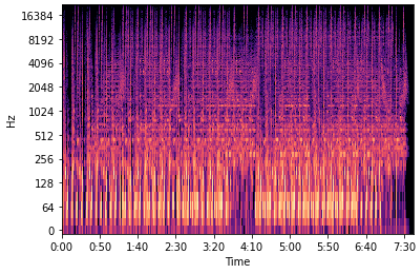


Ilustración 10 - espectrograma de una canción de muestra

4.3.3. Preparación de datos

De acuerdo con la definición que se dio en el entendimiento de negocio el primer paso que se realiza, es separar la canción en secciones de 15 segundos:

```
complete_filename = f'{BASE}/{filename}'
audio_data, sr = read_audio_file(complete_filename, 44100)
section_generator = get_section_from_audio(audio_data,sr,15)
for section, part in section_generator:

    temp_filename = filename[:-3].replace(".", "") + '_' + part + '.wav'
    #generar imagen de espectograma (futuros análisis)
    generate_spectogram_image(temp_filename, section, 'spectograms', './assets')
```

Ilustración 11 - separación de audio en secciones de 15 segundos

La función *get_section_from_audio*$(x(t), \hat{t})$ separa la onda en partes de 15 segundos de cada una de estas secciones, se calcularon características generales de la canción y se extraen las siguientes variables [16]:

- **Amplitude Envelope:** Es una característica del dominio del tiempo y se entiende como el valor de amplitud máxima dentro de un marco(frame). Está relacionado con el timbre de un sonido a lo largo del tiempo, es una propiedad que nos permite distinguirlo de otros sonidos.
- **Root mean square energy**: Una característica del domino del tiempo, toma la energía de todos los elementos de cada marco(frame) y calcula la raíz cuadrática media de la energía, se utiliza mucho en segmentación de audio, también puede ser utilizada para determinar silencios en una señal de audio.

$$\sqrt{\frac{1}{N}\sum_{n}|x(n)|^2}$$

- **Chroma Short Term Fourier Transform**: Puede identificar amplitudes de onda en una señal de audio. Muy utilizado para identificar notas musicales porque es robusta a los cambios de timbre.
- **Spectral centroid**: Se define como el centroide de cada barra de un espectograma normalizado
- **Spectral Bandwith**: margen superior e inferior del *spectral centroid* ejemplo:

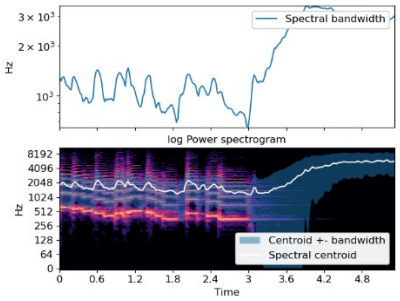


Ilustración 12 - Spectral bandwith

- Rolloff: se define para marco (frame) como el centro de frecuencia de un espectograma en el cual se encuentre el 85% de los valores se encuentren por debajo de él:

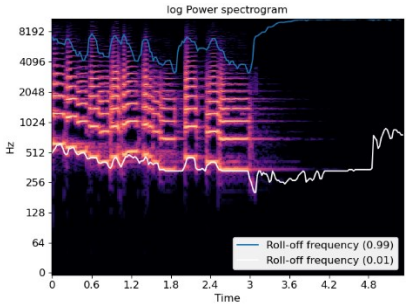


Ilustración 13 - Spectral rolloff

- Zero Crossing Rate: indica cuantas veces pasa una señal de audio por el valor 0, es muy útil para identificar sonidos de percusión o ruido, porque no son sonidos sostenidos. Se usa mucho para estimar sonidos monofónicos.
- Mfcc's: Se extraen 20 coeficientes para el análisis.

Cabe aclarar que se extraen muchas variables, pero no son utilizadas todas en el análisis

Para evitar la pérdida de información que puede ocasionar aplicar la función de Hann[2] a una ventana de audio y robustecer la extracción de características, se definen dos medidas: **FRAME_SIZE** y **HOP_SIZE**. La primera se encarga en dividir cada sección de 15 segundos en bloques (frames) más pequeños y el tamaño de salto (hop size) indica cuantos bits se debe regresar en la onda antes de calcular el siguiente marco (frame). Visualmente es lo siguiente:

[2] Es una función de ventana, que se utiliza para suavizar valores. https://en.wikipedia.org/wiki/Hann_function

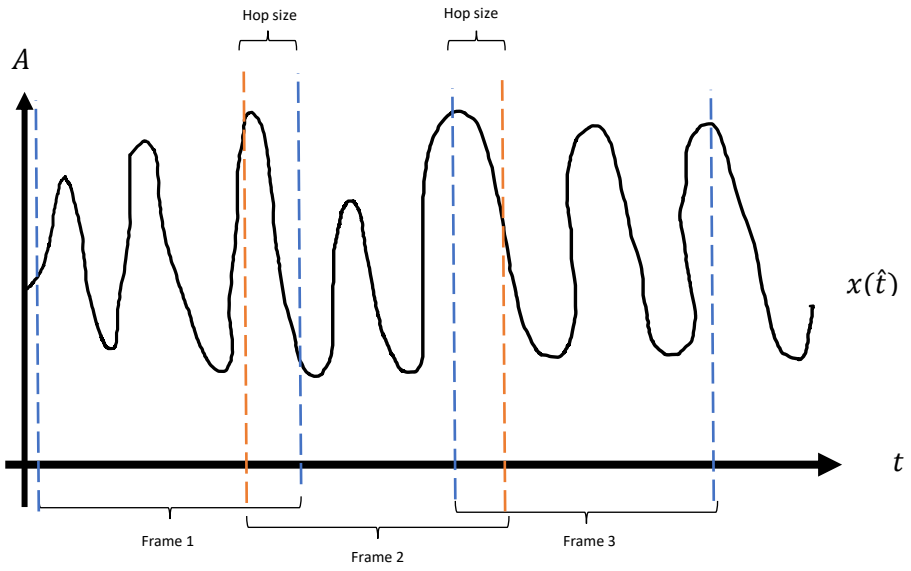


Ilustración 14 - Frame Size y Hop Size

Es bueno realizar esta operación porque en la implementación de los algoritmos para obtener una transformación de Fourier, en los límites de los marcos (frames) se pierde información o se crean valores erróneos por división con valores muy cercanos a cero. Para impedir esto se aplica una función de Hann por defecto[3].

Para obtener una tabla que indica por canción y parte cada una de estas variables

filename	part	amplitude_envelope	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	rolloff	...	mfcc17	mfcc18	mfcc19	mfcc20	la
Death on the Balcony - Tempt Of Fate.wav	part_5	0.822617	0.242730	0.389962	4207.524443	4478.295692	9054.711914	...	-3.063989	6.156282	4.394158	6.748927	
Death on the Balcony - Tempt Of Fate.wav	part_10	0.841110	0.246409	0.394201	3998.448896	4457.312863	8812.463379	...	-2.447684	5.184586	0.994026	3.922777	

Ilustración 15 - variables extraídas para cada sección de audio

El siguiente paso de preparación de datos es obtener las secciones donde hay mayor concentración de energía, porque estas contienen más información sobre los instrumentos y tempos del género musical [3]. Se normalizan estos datos, para poder establecer una relación entre ellos.

Primero se mide la concentración de energía como una razón entre las variables

- **Root Mean Square Energy:** Toma las ventanas de tiempo y extrae las secciones que contienen mayor energía para cada muestra de `n` segundos que se le pasa como parámetro a la función `get_section_from_audio` recordar que se definen un tamaño de marco(frame) y de salto(hop) por este motivo el proceso de cálculo es un poco más lento, pero más robusto

[3] Esta operación se llama windowing

- **Spectral centroid:** Se toman los marcos(frames) de la secuencia de audio y obtiene las frecuencias más importantes de la secuencia de audio, se espera que las secciones que tengan mayor volumen sean donde se encuentren las secciones más importantes de la canción.

De esta manera establecemos una relación entre una característica del dominio del tiempo y otra característica del dominio de la frecuencia, con esta separación para cada canción se pudo **determinar de manera exitosa cuales partes o secciones son más representativas de la canción.**

Ilustración 16 - Selección partes centrales de una canción

4.3.3.1. Selección de características

Después de varias iteraciones y revisar literatura sobre las variables más utilizadas sobre las características para clasificación se alcanzaron a las siguientes conclusiones:

- Solamente se selecciona el *spectral centroid* para el análisis dado que los otros valores espectrales ya se encuentran incluidos en este por definición.
- La literatura recomienda utilizar entre **12 y 14 coeficientes de MFCC**, incrementarlos no incrementa la calidad del modelo y en caso de querer mejorar la precisión de los coeficientes se recomienda utilizar la derivada en vez de aumentar la cantidad de coeficientes. Por este motivo solo se seleccionaron los 13 primeros coeficientes de los 20 extraídos.
- Zero crossing rate: Porque en este género musical hay patrones rítmicos con mucha percusión, y esta variable detecta precisamente esos patrones.
- Amplitude envelope: Porque da una idea del volumen dentro de la canción por lo que a priori podría ser un buen candidato.
- RMSE Root Mean Square Energy: Se usa mucho en clasificación para determinar la energía en cada ventana de audio.

Para un total **de 17 características.**

4.3.3.2. Pasos adicionales

Una vez realizada la selección de características se realizaron los siguientes pasos de preparación de datos:

- Se verifica que no exista una correlación alta entre las variables seleccionadas.
- Se aplica normalización de los datos.
- Se eliminan registros duplicados: Los registros duplicados encontrados no son inherentes a la data. Estos ocurren porque son canciones con patrones rítmicos muy marcados y porque se utilizaron canciones con versiones similares, dado que entre los datos hay versiones diferentes de la misma canción.
- Se verifica que la diferencia entre la media y la mediana para las características extraídas no sean significativas.

4.3.4. Modelado (K-Means variables clásicas)

Para este primer modelo se optó por un modelo de clasificación no supervisado llamado K-Means para determinar el número de K o grupos se utilizo la gráfica del codo, este método es útil como primer acercamiento porque la data no está preclasificada ni etiquetada.

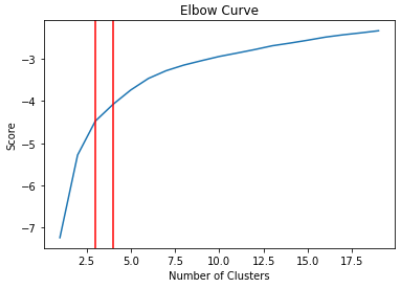


Ilustración 17 - Curva del codo

Dado este resultado el número de grupos debería estar entre **3 y 4**, aunque el número recomendable de grupos dado por el algoritmo podría ajustarse a 3, se optó por **4** grupos porque permite un análisis más detallado de los grupos similares y que características tienen en común.

Después de aplicar la clasificación se obtuvieron los siguientes centroides:

Out[30]:

	amplitude_envelope	rmse	zero_crossing_rate	spectral_centroid	mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	mfcc6	mfcc7	mfcc8
0	0.033903	0.035507	0.025878	0.030867	-0.033359	0.035625	-0.017614	0.034622	-0.012330	0.032727	0.011374	0.029422
1	0.040839	0.035928	0.025276	0.037155	-0.033266	0.026125	-0.026076	0.043759	0.014190	0.047685	0.050294	0.047994
2	0.026307	0.032084	0.033321	0.034511	-0.031694	0.034183	-0.017034	0.031784	-0.030844	0.032120	-0.026261	0.031790
3	0.031673	0.031221	0.050226	0.038689	-0.031321	0.030897	-0.037703	0.021854	-0.051841	0.017425	-0.047754	0.000258

mfcc9	mfcc10	mfcc11	mfcc12	mfcc13
0.017658	0.029290	0.016675	0.028592	0.016077
0.058466	0.050195	0.065433	0.046710	0.040643
-0.017356	0.027108	-0.013684	0.029565	-0.007380
-0.053814	-0.012768	-0.051570	-0.013616	-0.057148

Ilustración 18 - centroides

Se utilizó una métrica euclídea y al ser datos del mismo género musical con ritmos y tempos muy similares las secciones son muy cercanas las unas a las otras y no se alcanzan a observar grupos claramente definidos.

Para observar este fenómeno se aplicó PCA a la data y a los centroides para reducir su dimensionalidad y observar resultados en dos dimensiones, la explicabilidad obtenida fue la siguiente:

$$pca(X) = [0.4894 \quad 0.1286] \approx 60\%$$

$$pca(C) = [0.9592 \quad 0.034] \approx 98\%$$

y la gráfica obtenida fue la siguiente:

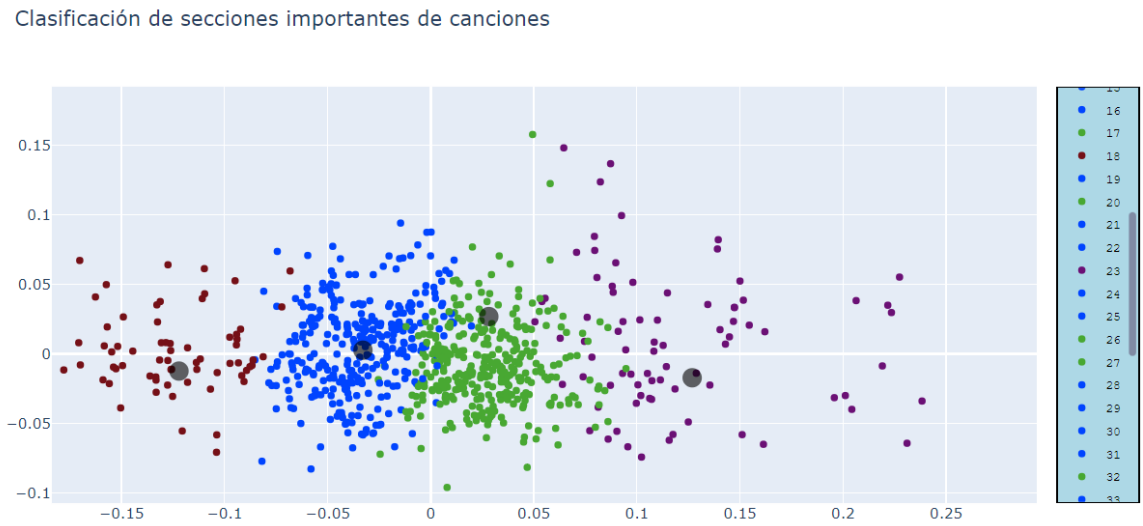


Ilustración 19 - Grupos K-Means Variables clásicas

Donde cada punto representa una sección de canción de 15 segundos. Se observa que los grupos no están bien definidos. Desde el punto de vista del análisis de datos. **Sin embargo, en la fase de evaluación arrojó unos datos interesantes que se mostrarán a continuación.**

4.3.5. Evaluación (Variables Clásicas)

Dados los resultados del punto anterior se aplicó análisis de silueta al algoritmo de K-Means que confirmó las sospechas iniciales del paso de modelado, ***0.17832824073167858*** que indica que algunos grupos se encuentran superpuestos.

También se realiza un análisis para identificar la incidencia de cada variable en el grupo, este se realizó utilizando primero utilizando un árbol de decisión utilizando como variable objetivo las etiquetas generadas por el algoritmo de K-Means, obteniendo el siguiente resultado:

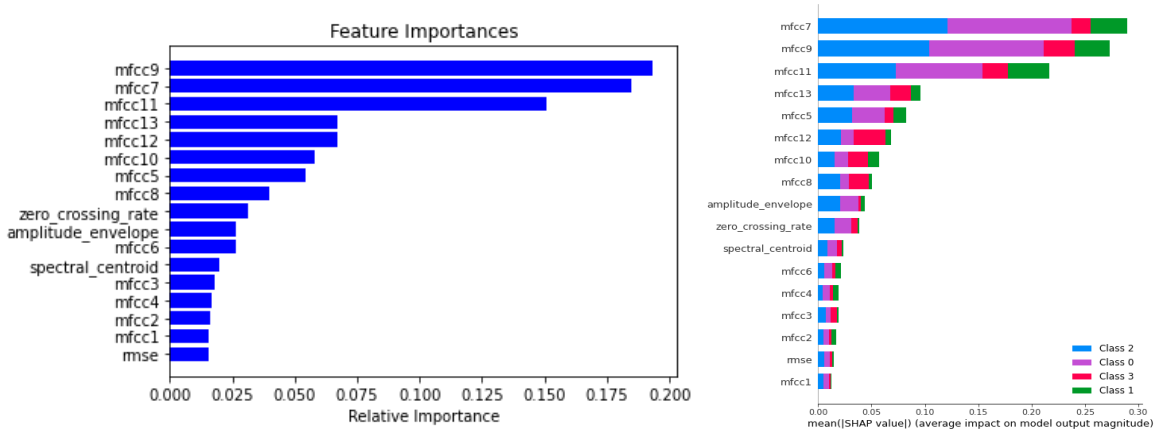


Ilustración 20 - resultado variables más importantes árboles de decisión

La ilustración anterior muestra el resultado aplicando dos métodos distintos el de la izquierda es un *RandomForestClassifier* y a la derecha aplicando un análisis SHAP [17], los resultados, aunque no son iguales muestran una consistencia entre los dos métodos e indica que las características: *mfcc7, mfcc9 y mfcc11, **son las que más influyen en la selección del clúster***.

De hecho, el análisis SHAP permite determinar cómo influye en cada clúster la característica. Mostrando que para el clúster 3 el mfcc7 no es tan importante.

Sin embargo, aunque el modelo parece no haber generado buenos grupos debido a la naturaleza de los datos, **al analizar los puntos en la gráfica y realizar una comparación auditiva entre los clústeres si se nota una diferencia entre ellos, aunque teóricamente se encuentren demasiado juntos para que exista una diferencia.**

Este hallazgo anima a implementar otras metodologías y estrategias para mostrar la diferencia (o cercanía entre las secciones de audio), entre las estrategias seleccionadas tenemos:

- **Implementar un tablero de control** donde se pueda interactuar de manera más dinámica entre el clúster y la canción, facilitar la interpretabilidad de los resultados del modelo.
- **Implementar un modelo de K-Means** Donde la selección de características sea a partir de la información del espectograma de la canción, pues este contiene más información para una red neuronal que los mfcc.

Con estas dos estrategias en mente se continua con el desarrollo del proyecto.

4.4. PROCEDIMIENTO DE CLASIFICACIÓN UTILIZANDO EXTRACCIÓN DE CARACTERÍSTICAS UTILIZANDO REDES NEURONALES

Explica el procedimiento realizado en Google Collab: ***Clasificación de canciones utilizando red neuronal para selección de características.ipynb.*** Disponible en el siguiente enlace: https://colab.research.google.com/drive/1PVvmKLSLgLXBRtV0IIm0X20wiukK6Kra?usp=sharing

4.4.1. Entendimiento de negocio

En el paso anterior fueron extraídas para cada canción sus secciones centrales, que son las secciones donde hay una mayor concentración de energía y fuerza, con base a esto se implementó el modelo de K-Means. Con el objetivo de tener resultados comparables para esta sección se toma como insumo esta misma de base de segmentos de 15 segundos.

Para obtener las características de cada sección se utiliza una red neuronal convolucional pre-entrenada, que es un puerto[4] del siguiente artículo [18], en este se encontró que las arquitecturas de redes neuronales modernas para clasificación de imágenes pueden ser muy eficientes para el procesamiento de audio. Este trabajo fue adaptado por Harry Taylor en su librería *torchvggish*.

Este modelo es capaz de extraer 128 características por **cada segundo** de audio como se muestra en la estructura de la red:

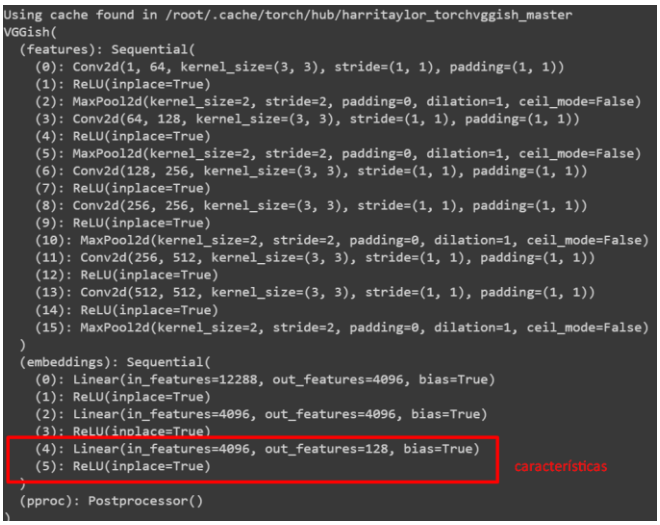


Ilustración 21 - Estructura red CNN Vggish

Por este motivo se opta por un acercamiento que particiona el segmento original de audio de 15 segundos en secciones de 1 segundo.

[4] Un puerto es una implementación en Python de una librería existente en otro lenguaje de programación. Es una traducción del término *port*

4.4.2. Preparación de Datos

Del punto anterior se realiza un procedimiento que genera un sistema de archivos, que contempla una carpeta con el nombre de la canción, y un archivo (.WAV) cuyo nombre es la parte de la canción concatenada la sección que constituye el segundo

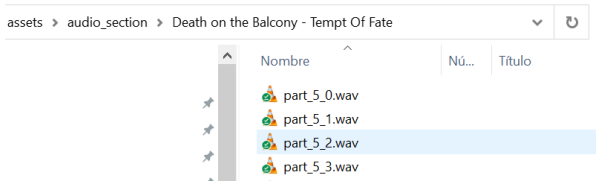


Ilustración 22 - Nombre canción sección y parte

De manera análoga se calculan sus respectivos espectrogramas:

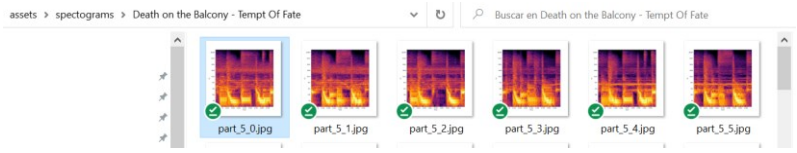


Ilustración 23 - espectograma secciones de audio

De este proceso se obtiene un dataframe con 128 características cuyos valores están en un rango de *[0, 255]*:

	filename	section	part	f0	f1	f2	f3	f4	f5	f6	...	f118	f119	f120	f121	f122	f123	f124	f125	f126	f127
10634	v_Urban Blues Project pres Michael Procter - L...	0	part_3	175.0	10.0	160.0	84.0	197.0	111.0	115.0	...	0.0	75.0	71.0	255.0	113.0	74.0	255.0	0.0	54.0	255.0
10635	v_Urban Blues Project pres Michael Procter - L...	1	part_3	170.0	18.0	164.0	103.0	181.0	68.0	76.0	...	0.0	89.0	122.0	190.0	78.0	88.0	247.0	79.0	0.0	255.0

Ilustración 24 - data preparada

Estas 128 características representan la información extraída del espectograma de la canción.

Se eliminan registros duplicados de 1 segundo, no es necesaria una normalización porque todos los datos se encuentran en la misma escala y se procede con la sección de Modelado.

4.4.3. Modelado (K-Means características VGGish)

De la clasificación anterior y para determinar si estas características generan mejores resultados de que las variables clásicas se repite el acercamiento con K-Means, cuya curva del codo fue la siguiente:

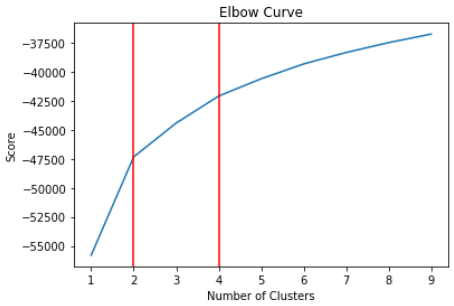


Ilustración 25 - Curva del codo

Que muestra dos posibles puntos K= [2,4] de nuevo para mantener la idea del punto de entendimiento de negocios y que los resultados del modelo sean comparables se opta por un **K=4**, se mantiene la distancia euclídea y se grafican resultados utilizando PCA para reducir la dimensionalidad de los grupos con los siguientes resultados.

$$pca(X) = [0.2089322 \quad 0.08762488] \approx 29\%$$

$$pca(C) = [0.6938393 \quad 0.17371095] \approx 90\%$$

Algo que demuestra que la gráfica en dos dimensiones no es tan confiable, pero puede ayudar a explicar la naturaleza de los clústeres generados:

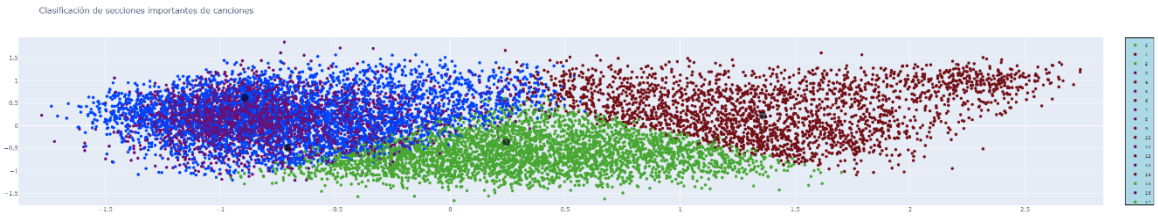


Ilustración 26 - Grupos K-Means características VGGish

Se observan tres grupos bien definidos (azul, granate y verde), pero dos grupos se superponen, porque sus características son similares en R^2 pero más definidas en R^{128} se utiliza SHAP análisis para determinar las características más influyentes en el modelo, pero su explicabilidad genera un inconveniente para la evaluación por este motivo se procede con una metodología de evaluación distinta.

4.4.4. **Evaluación (Características CNN)**

Para evaluar los resultados del modelo se opta por una metodología distinta, que requiere conocer cuáles son las características que se activan en una capa específica de la red neuronal (generalmente en la última capa). Este algoritmo se conoce como Grad-CAM [19]. Se genera para cada sección de audio una imagen como la siguiente:

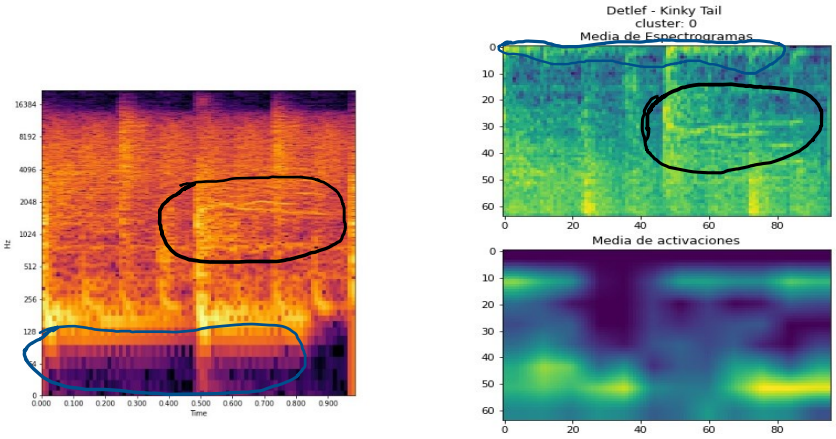


Ilustración 27 - Espectograma y media de activación sección 7 parte 7 de la canción Detlef

En la ilustración se observan las siguientes características:

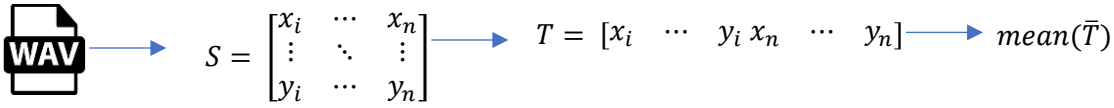
- Los espectrogramas, aunque generados con técnicas distintas (librosa, vggish) coinciden.
- Debido al proceso interno que realizan algunas capas del modelo VGGish el espectograma de la derecha está invertido.
- El círculo dibujado de color negro muestra una sección de la canción que es representativa.
- Los ejes '*y*' de las gráficas no coinciden porque uno es generado durante una capa de convolución que va hasta 64, como se observa en esta Ilustración 21 - Estructura red CNN VggishIlustración 21 mientras que el de la izquierda está en una escala logarítmica porque facilita la interpretabilidad por parte de un humano.
- El círculo dibujado de color azul muestra una zona que debido a la escala del espectograma no se alcanza a observar, sin embargo, se encuentra presente en ambos espectrogramas.
- La gráfica de "Media de activaciones" indica cuales fueron las capas que el algoritmo de Grad-CAM [19] activo antes de generar el resultado final que son las 128 características. Es decir que existe una función que mapea 1 a 1 esta gráfica con las 128 características.

Con estos resultados y dada la clasificación que se obtuvo del algoritmo de K-Means podemos relacionar, las capas medias de activación con los centroides generados por el algoritmo. Utilizando este pseudocódigo:

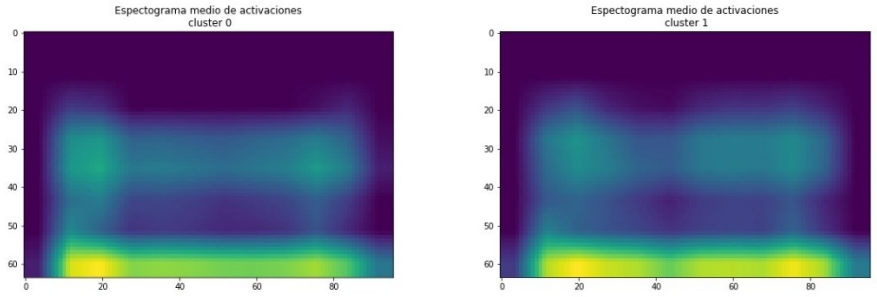
```
Para_cada k en clusters:
  arreglo_k = []
  Para_cada seg1 en el total_de_datos:
    spectrogram = calcular_spectograma_gradcam(seg1)
    gray = obtener_media_de_activacion_gradcam(spectrogram)
    arreglo_k.insertar(gray)
  activacion_media_cluster = media(arreglo_k)
  generar_grafica(activacion_media_cluster)
```

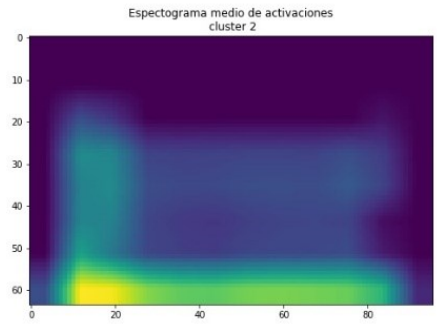
Donde *seg1* es el segmento de canción de 1 segundo, La media o la mediana pueden ser utilizadas para obtener el resultado en la línea 7. **Todo el procedimiento fue realizado utilizando tensores en un ambiente de Google Collab.**

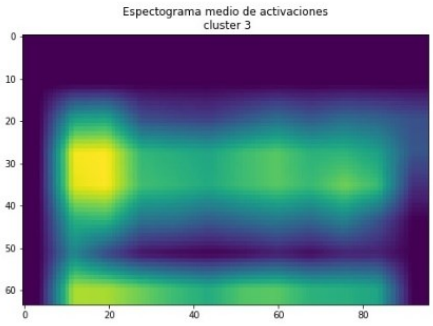
Intuitivamente el proceso que se realiza para cada sección de audio es el siguiente:



Esto arroja los siguientes resultados de activación media por cada clúster:







Los clústeres que se encuentran superpuestos o muy similares son el **0** y el **1.** Obtenemos esta conclusión al observar los espectrogramas medios de activación

Sin embargo, se realiza otro análisis para poder visualizar data de alta dimensionalidad dado que el algoritmo de PCA pierde mucha información, que solo alcanza a explicar una varianza del 30%, para ello se generan los archivos:

- t-sne-input-data-without-metadada.tsv
- t-sne-input-data-metadada.tsv

Estos archivos contienen la data cruda después de la visualización, sirven de insumo para el algoritmo T-SNE [20] y permiten replicar los resultados de este documento, se puede replicar este ejercicio en el siguiente enlace y cargar los archivos t-SNE generados.

Una vez cargados los datos se genera un gráfico T-SNE con los parámetros de la Ilustración 28, cuya ejecución genera la gráfica Ilustración 29. Estos resultados se analizan en cada una de las secciones siguientes:

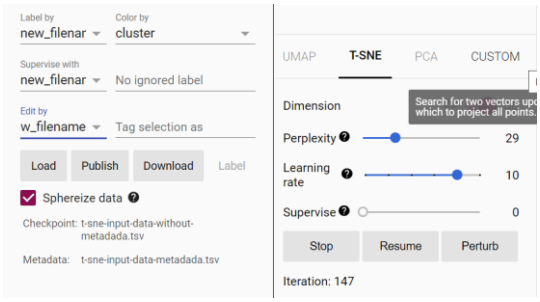


Ilustración 28 - configuración para gráfico T-SNE

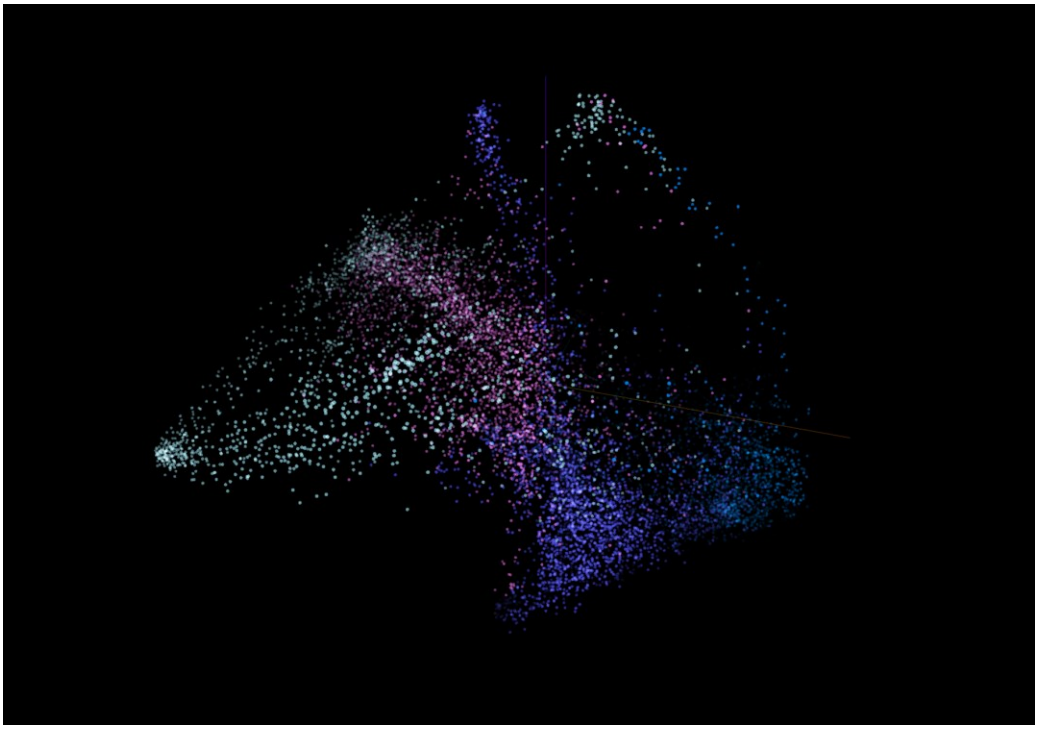


Ilustración 29 - Resultado algoritmo T-SNE

4.4.4.1. Resultados T-SNE puntos centrales

Una ventaja del algoritmo T-SNE es que puede detectar que puntos se encuentran relacionados, aunque su distancia euclídea no sea la menor, con esto en mente se realiza un acercamiento al punto central de la gráfica y se obtiene que los segmentos más relacionados se muestran en la

Ilustración 30, con esta información se genera una lista de reproducción adjunta en el repositorio del proyecto denominada "**tsne-center-segments.xspf**"[5]. En el cual se incluyen los siguientes segmentos que si se encuentran relacionados:

- V-PSB – Before (12RDDJ 6431 C2)/part_14_0.wav
- V-PSB – Before (12RDDJ 6431 C2)/part_14_1.wav
- V-PSB – Before (12RDDJ 6431 C2)/part_13_15.wav
- v_Urban Blues Project pres. Michael Procter - Love Don't Live (Soulfuruc Dub)/part_10_10.wav
- v_Urban Blues Project pres. Michael Procter - Love Don't Live (Soulfuruc Dub)/part_10_9.wav
- v_Full Intention - America (I Love America) (Nevins Goldfinger Mix)/part_22_0.wav

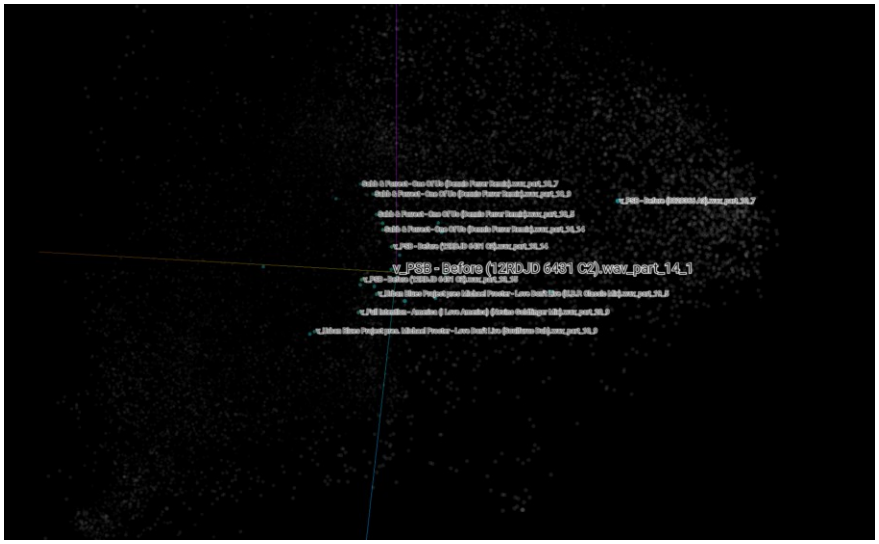


Ilustración 30 - T-SNE puntos centrales

En el análisis de puntos centrales existe una relación entre los segmentos, pero estos pueden pertenecer a distintos clústeres.

4.4.4.2. Resultados T-SNE puntos centrales

En este resultado se repite este análisis con segmentos de canciones que se encuentran en los límites de la nube de puntos:

[5] Este tipo de archivos se pueden abrir con un reproductor VLC https://www.videolan.org/vlc/index.es.html

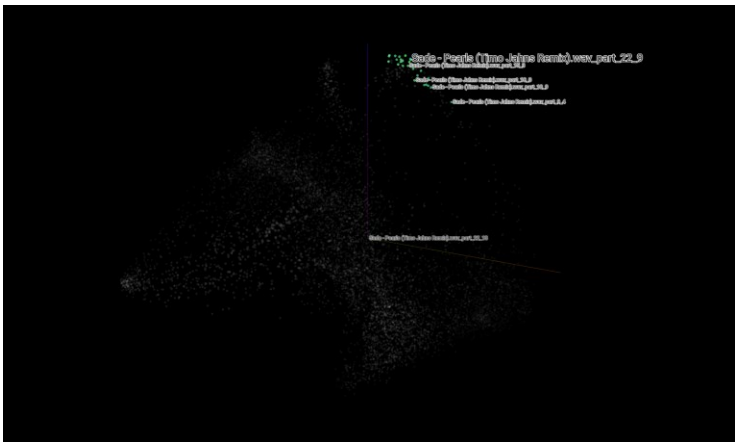


Ilustración 31 - T-SNE puntos limítrofes

Repitiendo procedimiento anterior se genera una lista de reproducción con los puntos relacionados llamado "**tsne-related-segments.xspf**":

- Sade - Pearls (Timo Jahns Remix)/part_22_9.wav
- Sade - Pearls (Timo Jahns Remix)/part_22_10.wav
- Sade - Pearls (Timo Jahns Remix)/part_14_6.wav
- Sade - Pearls (Timo Jahns Remix)/part_18_2.wav
- Sade - Pearls (Timo Jahns Remix)/part_10_5.wav
- Siopis ft Metrika - Linda (Lemos & Pan Remix)/part_12_0.wav

Estos resultados muestran que el algoritmo es capaz de inferir (en algunos casos) de manera autónoma la información de las características que se obtienen de la red neuronal que segmentos se encuentran muy relacionados.

4.4.4.3. Segmentos perdidos

El último análisis que se realiza es sobre una nube de puntos anómala perteneciente a un clúster pero claramente alejada de la nube de puntos principal, como se muestra en la Ilustración 32

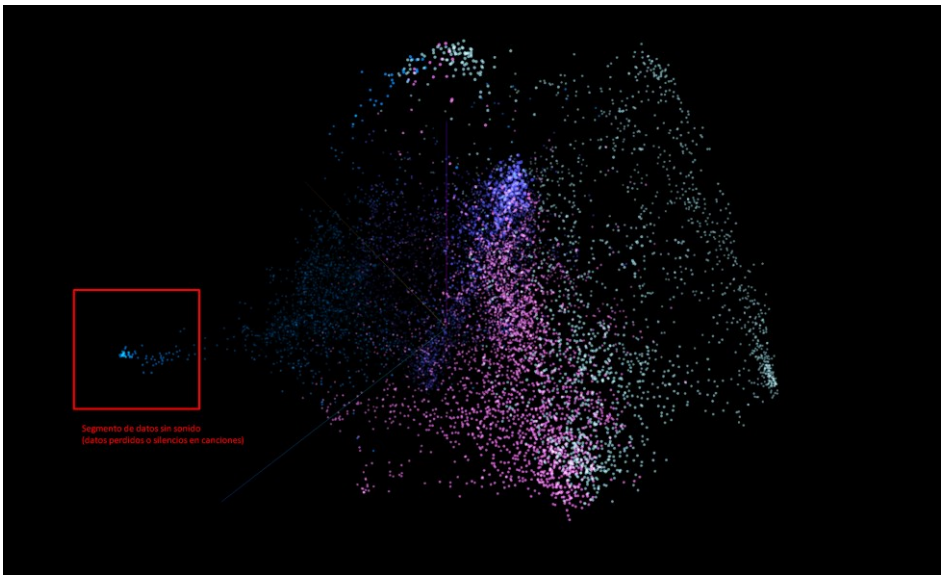


Ilustración 32 - T-SNE segmentos anómalos

Al generar una lista de reproducción denominada "**tsne-missing-segments-or-silence-songs.xspf**" con los siguientes segmentos:

- v_Full Intention - America (I Love America) (_Rude Dog_ Mix)/part_20_10.wav
- v_Full Intention - America (I Love America) (_Rude Dog_ Mix)/part_22_10.wav
- v_Full Intention - America (I Love America) (_Rude Dog_ Mix)/part_22_3.wav
- v_Full Intention - America (I Love America) (Full Length 12_ Vocal Mix)/part_23_15.wav
- v_PSB - Before (8828366 A3)/part_27_14.wav
- v_PSB - Before (8828366 B1)/part_28_14.wav

Estos resultados pueden deberse al método de extracción del segmento o a silencios existentes dentro de una canción, pero de nuevo prueban la efectividad de este algoritmo para entender esta data de alta dimensionalidad.

5. EVALUACIÓN

Al ser K-Means un método de clasificación no supervisado y dada la naturaleza similar de los datos, crear grupos se torna complejo a la hora de validar los resultados. Sin embargo, a pesar de que las metodologías de validación del algoritmo con variables clásicas parecen no ser satisfactorias, durante la escucha de los segmentos si existen grupos definidos.

En las secciones **Modelado (K-Means variables clásicas)** y **Evaluación (Características CNN)**, se muestran los resultados del proceso de modelado utilizando distintos procedimientos, pero el objetivo de generar el acercamiento con ambos modelos es que los resultados puedan ser comparables para lograr esto se debe realizar un pequeño ajuste sobre el resultado de CNN.

El ajuste consiste en poder comparar segmentos de igual duración que originalmente serían 15 segundos, luego para poder comparar los resultados de ambos modelos se aplica una *moda* para cada sección y el valor que más se repite por sección, será el valor del clúster para el algoritmo K-Means con CNN.

Al realizar este ajuste ya se tienen dos resultados comparables, sin embargo, la última validación se debe **realizar a juicio de expertos**, para ayudar con el proceso de clasificación se realiza un tablero de control, que ayuda con la exploración de resultados. Adjunto en el repositorio del proyecto.

5.1. TABLERO DE CONTROL

Es un proyecto realizado con las librerías Plotly, Dash en el cual se puede realizar una exploración de los audios presentes en el proyecto, cuenta con tres páginas principales:

5.1.1. Vista Principal (características clásicas)

Resume el resultado del proceso de clasificación utilizando variables clásicas en la primera sección, se encuentran los controles de análisis del proyecto, como se muestra en la Ilustración 33 - Sección de controles variables clásicas. Al presionar el botón actualizar se modifican las tres pestañas de la columna 1 o 2.

La primera pestaña indica de las secciones centrales de la canción en qué clústeres se encuentra distribuida, la segunda pestaña permite escuchar la sección que se encuentre marcada en la lista desplegable “Seleccionar Secciones Canción” o todas si no hay ninguna parte marcada y la tercera pestaña permite obtener una vista detallada (tabla) de las partes seleccionadas.

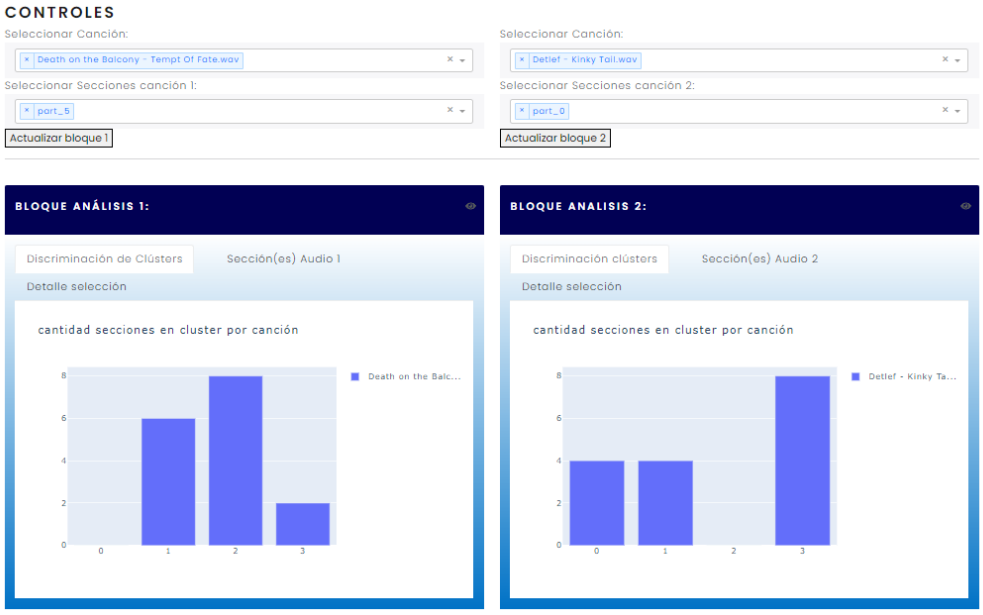


Ilustración 33 - Sección de controles variables clásicas

En la segunda columna permite comparar los segmentos de manera espacial con una vista gráfica:

Ilustración 34 - Análisis exploratorio por sección

Al seleccionar un punto de la gráfica carga la información del espectograma asociado y de la sección de canción mostrada, también se muestra un análisis SHAP para saber la influencia una variable en la elección del clúster.

5.1.2. Vista (Redes Neuronales)

Permite explorar los clústeres generados utilizando la selección de características del modelo VGG, para el primer análisis solo se cargan las imágenes generales por clúster.

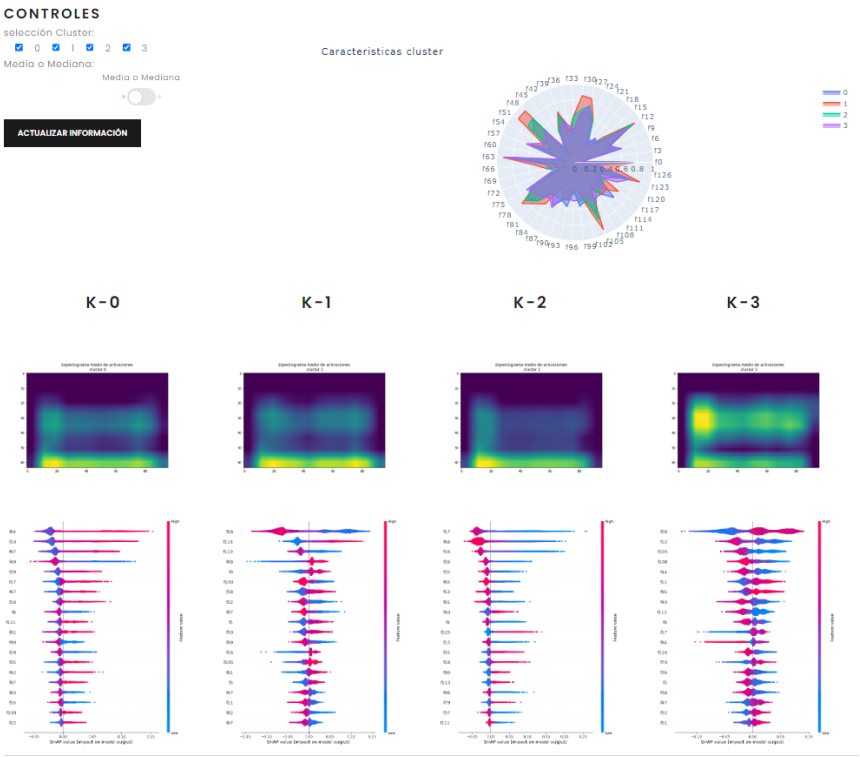
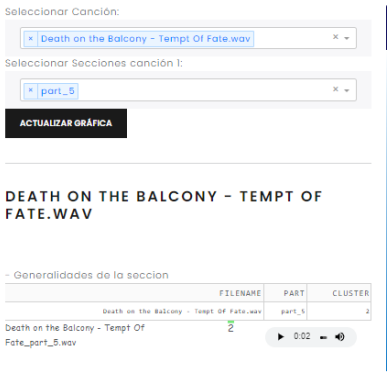


Ilustración 35 - Sección de resultados general

En la sección inferior se puede realizar un análisis exploratorio similar al de la vista Ilustración 34,

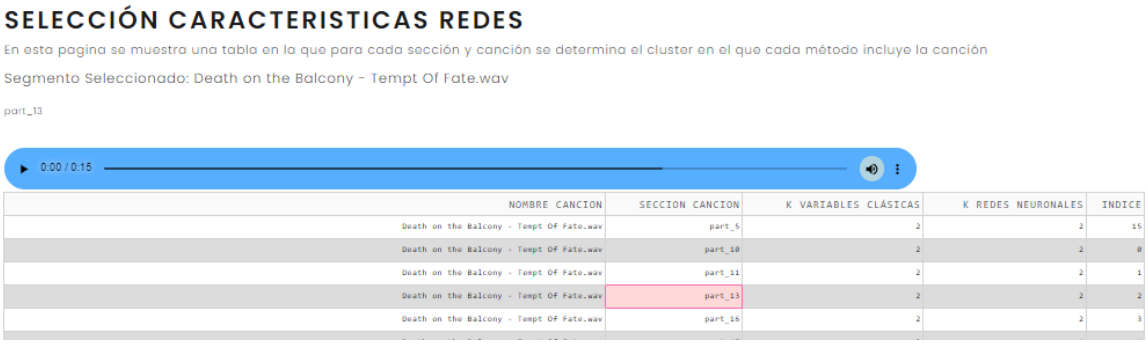
Ilustración 36 - Análisis exploratorio de segmentos de 1 segundo

Al seleccionar un punto de la gráfica se carga el detalle en un cuadro y la sección para poder escuchar la sección como se muestra en la siguiente imagen:



5.1.3. Vista (Comparación métodos)

En esta vista se encuentra una tabla que muestra la comparación del clúster asignado sobre cada método y es sobre este resultado que se aplica un juicio de expertos de los resultados obtenidos por los modelos.



Al seleccionar una fila se carga la sección de canción solicitada y se puede determinar si el clúster asignado es correcto o no.

6. DESPLIEGUE

El tablero de control fue desplegado en un ambiente de AWS y se encuentra disponible en el siguiente enlace: http://184.73.62.159/

Los datos de ingreso son:

Usuario: proyecto_clasificacion

Contraseña: eafit2022

Se despliega el tablero en una máquina de EC2 *small* con conexión a un *bucket* de S3, el proceso de despliegue fue descrito en detalle en el archivo README del repositorio, disponible en:

https://github.com/cabymetal/audio_analysis

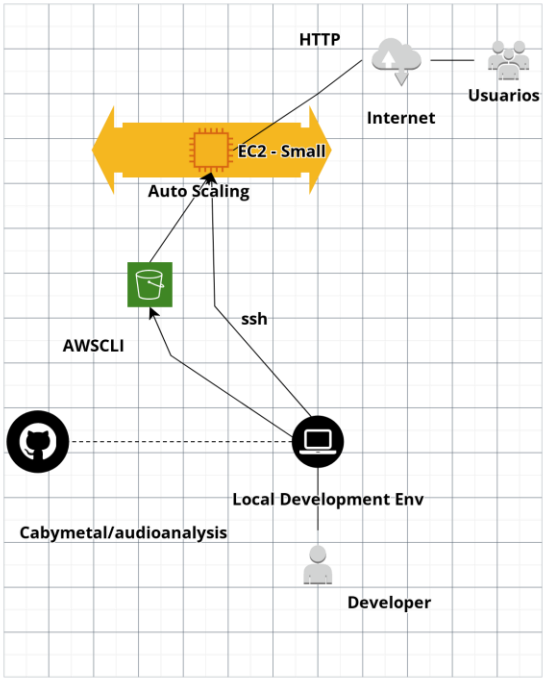


Ilustración 37 - Arquitectura AWS

La carpeta *assets* donde se encuentra toda la información relacionada con, espectrogramas segmentación de audio, e imágenes y data relacionada con este proyecto está publicada en una carpeta de drive pública en el siguiente enlace:

https://drive.google.com/drive/folders/1-YQv_Udzz4rWuRO5ttEDFLtBkqofqyc7?usp=sharing .

7. CONCLUSIONES Y TRABAJO FUTURO

El objetivo principal de este trabajo es poder determinar distancia entre segmentos de canciones el cual se cumplió utilizando distintas formas de selección de características, por este motivo el trabajo futuro además de estar relacionado con aplicar los resultados en la creación del dj automático [3], también se encuentra relacionado con la verificación de estos resultados aplicando juicio de expertos.

La clasificación de audio, utilizando K-Means ha mostrado ser esperanzadora porque las características seleccionadas utilizando ambas metodologías han sido probadas en diferentes ámbitos con buenos resultados. Además, el modelo VGG ha logrado identificar patrones en las canciones que con variables clásicas se hubieran tenido que preprocesar, por ejemplo: detección de silencios.

Los silencios existen en la muestra porque no se realiza ningún tipo de preprocesamiento de la canción tomando una ventana de tiempo pequeña en la escala de milisegundos. Es decir que en un segmento de 1 segundo puede existir mucha concentración de energía, pero puede existir un porcentaje x de silencios en la escala de milisegundos.

En este proyecto se utilizaron otros métodos de clasificación como soft clustering y Knn, pero los resultados no fueron satisfactorios dada la naturaleza de los datos muy similar, y que requieren un proceso de etiquetado inicial con el cual no se contaba.

Otro punto interesante para evaluar es que se establece una metodología para caracterización de segmentos musicales, pero mucho del desarrollo se utilizó utilizando una métrica euclídea sería interesante, reemplazar esta métrica por otra y comparar sus resultados gráficamente. Para identificar si se mejora la explicabilidad de las gráficas con los resultados auditivos obtenidos.

A pesar de que no se obtuvieron clúster claramente definidos gráficamente, ambas metodologías arrojaron resultados que muestran una clasificación de los segmentos de canciones con grupos definidos de forma auditiva, los elementos de las fronteras pueden estar relacionados o pertenecer a varios clústeres a la vez, en este punto es donde una metodología una clasificación de soft-clustering sería interesante. Y en un trabajo futuro sería bueno aplicarla y comparar sus resultados con el resultado de K-Means clásico.

También hay que mencionar que se podría mejorar la explicabilidad de algunas características, pero hacerlo requiere obtener más datos etiquetarlos y evaluar de nuevo la efectividad para reentrenar el modelo.

8. BIBLIOGRAFIA

[1] H. S. Alar, R. O. Mamaril, L. P. Villegas y J. R. D. Cabarrubias, «Audio classification of violin bowing techniques: An aid for beginners,» *Machine Learning with Applications,* p. 100028, 2021.

[2] H. Malik, U. Bashir y A. Ahmad, «Multi-classification neural network model for detection of abnormal heartbeat audio signals,» *Biomedical Engineering Advances,* p. 100048, 2022.

[3] V. Tideman, *Organization of Electronic Dance Music by,* 2022, p. 48.

[4] P. Knees y M. Schedl, Music Similarity and retrieval: an introduction to audio and web-based strategies, Berlin: Springer, 2016.

[5] M. Stéphane, A Wavelet Tour of Signal Processing (Third Edition), M. Stéphane, Ed., Academic Press, 2009, pp. 1-31, 481-533.

[6] D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano y J. Gonzalez-Rodriguez, «Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset,» *EURASIP Journal on Audio, Speech, and Music Processing,* vol. 2019, nº 9.

[7] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev y N. Chervyakov, «Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,» *Mathematics and Computers in Simulation,* p. 232–243, 2020.

[8] D. Debianux, Artist, *Espectograma 3d.* [Art].

[9] Anon, Artist, *espectograma violín.* [Art].

[10] G. C. a. B. Han, «Improve K-means clustering for audio data by exploring a reasonable sampling rate,» *Seventh International Conference on Fuzzy Systems and Knowledge Discovery,* pp. 1639-1642, 2010.

[11] R. Thiruvengatanadhan, «Speech/Music Classification using MFCC and KNN,» *ISSN 0973-1873 Volume 13,* vol. 13, pp. 2449-2452, 2017.

[12] M. Sahidullah y G. Saha, «Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition,» *Speech and comunication,* p. 543–565. , 2012.

[13] L. Tessarini y . A. M. Frattini Fileti, «Audio signals and artificial neural networks for classification of plastic resins for recycling,» *Digital Chemical Engineering,* pp. 100059,, 2022.

[14] K. Simonyan y A. Zisserman, «VERY DEEP CONVOLUTIONAL FOR LARGE SCALE IMAGE RECOGNITION,» *aRXIV,* p. 14, 2014.

[15] M. Gaurav, K. K. Mohbey, A. Indian y S. Kumar, «Sentiment Analysis from Images using VGG19 based Transfer Learning Approach,» *International Conference on Industry Sciences and Computer Science Innovation,* vol. 204, pp. 411-418, 2022.

[16] A. Klapuri y M. Davy, Signal processing methods for music transcription, New York: Springer, 09 May 2006.

[17] S. M. Lundberg y S.-i. Lee, «A Unified Approach to Interpreting Model,» *arxiv,* p. 10, 2017.

[18] S. Hershey, S. Chaudhuri, S. a. E. D. P. a. G. J. F. a. J. y A. a. M. , IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans: IEEE Explore, 2017, pp. 131-135.

[19] R. R. Selvaraju, M. a. D. Cogswell, A. a. V. R. a. P. D. a. B. y D. , «Grad-cam: Visual explanations from deep networks via gradient-based localization,» *Proceedings of the IEEE international conference on computer vision,* pp. 618-626, 2019.

[20] M. a. V. Wattenberg, F. a. J. y I. , «How to Use t-SNE Effectively,» *Distill,* 2016.

[21] P. Christensson, «Sample Rate Definition,» Mayo 2015. [En línea]. Available: https://techterms.com/definition/sample_rate.